Un sujet pour l'épreuve orale
Simulation et comparaison d'intervalles de confiance

Présentation

Le texte proposé ci-après est conçu pour l'épreuve orale en filière BCPST.

L'exercice est construit de façon à ce que la difficulté soit progressive. Le sujet commence donc par des questions de « prise en main », volontairement simples et proches du cours, et met en place, très tôt, l'usage des compétences informatiques dans une perspective de simulation.

La question 2 peut amener le candidat à inventer une fonction auxiliaire simulant une loi de Bernoulli à partir du générateur de nombres pseudo-aléatoires, mais une telle démarche n'est pas imposée. En (2b) la réponse la plus simple consiste à employer la fonction Python `sum()`, mais une boucle convient tout aussi bien. En (2c) le calcul direct est évidemment possible, mais on peut aussi s'appuyer sur le fait qu'on traite un tirage de Bernoulli, ce qui rend identiques la somme des valeurs et la somme des carrés.

La question 3 est quasiment une question de cours (théorème central limite), et la question 4 est une simple exploitation des questions précédentes.

La question 6 est aussi une question de cours, tandis que la question 7 amène à vérifier si le concept même d'intervalle de confiance a été compris.

La question 8 laisse la place à beaucoup plus d'initiative, permettant aux candidats les plus aguerris de démontrer leurs compétences les plus fines dans différents domaines (simulation, représentations visuelles, etc.).

On rappelle que l'emploi d'une calculatrice ou d'un logiciel (fourni) est autorisé dans cette épreuve.

Les compétences mobilisées dans ce sujet sont essentiellement les suivantes :

- ▷ Engager une recherche, définir une stratégie : questions 2, 5.
- ▷ Représenter, changer de registre : question 5.
- ▷ Mobiliser des connaissances scientifiques pertinentes : questions 1, 3, 4, 6.
- ▷ Traduire un algorithme dans un langage de programmation : questions 2, 5.
- ▷ Critiquer ou valider un modèle ou un résultat : questions 7, 8.
- ▷ Argumenter, convaincre : compétence présente, par nature, dans l'ensemble de l'épreuve.
- ▷ Communiquer à l'écrit et à l'oral : compétence présente, par nature, dans l'ensemble de l'épreuve.

Énoncé

Dans une population d'individus amateurs de café, une proportion p (inconnue) préfère le robusta à l'arabica. On interroge n individus de cette population et on note $X_i = 1$ si l'individu i préfère le robusta et $X_i = 0$ sinon. On suppose que les X_i sont indépendantes. Soit Z_n le nombre d'individus interrogés préférant le robusta à l'arabica.

1. Quelle est la loi de Z_n ? Donner son espérance et sa variance.
2. On se propose de simuler informatiquement le tirage des X_i ainsi que les informations statistiques qu'on peut en tirer ; on rappelle à cet effet que la fonction `random()` de la bibliothèque Python `random` renvoie un nombre pseudo-aléatoire que l'on peut supposer uniformément distribué entre 0 et 1.

(a) Proposer une fonction Python `observation()` qui, pour n et p donnés en entrée, renvoie une liste de 0 et 1 correspondant aux valeurs prises par les X_i pour une observation d'un échantillon aléatoire de taille n répondant au schéma de Bernoulli de paramètre p.

(b) Proposer une fonction Python `moyempir()` fournissant la moyenne empirique (c'est-à-dire, la fréquence des 1) à partir de la donnée d'un échantillon sous la forme d'une liste de 0 et de 1.

(c) Proposer une fonction Python `varempir()` fournissant la variance empirique à partir de la donnée d'un échantillon sous la forme d'une liste de 0 et de 1.

Dans la suite, on souhaite proposer (par diverses méthodes) un intervalle de confiance pour p de niveau de confiance 99%, c'est-à-dire un intervalle que l'on peut calculer à partir des observations dont on dispose (c'est-à-dire, Z_n) et auquel p appartient pour plus de 99% des échantillons utilisés.

3. Montrer que pour n assez grand il existe un nombre u pour lequel

$$\mathbb{P}\left(\frac{Z_n}{n} - u\sqrt{\frac{p(1-p)}{n}} \leqslant p \leqslant \frac{Z_n}{n} + u\sqrt{\frac{p(1-p)}{n}}\right) \approx 0,99\,. \tag{1}$$

Pour accéder concrètement au nombre u, on pourra faire appel à la bibliothèque `scipy.stats` qui fournit les fonctions suivantes :
`norm.cdf()` qui donne la fonction de répartition d'une loi normale centrée réduite,
`norm.ppf()` qui donne la fonction réciproque de la précédente (également nommée *fonction des quantiles*).

4. Montrer que pour tout $p \in [0,1]$, $p(1-p) \leqslant 1/4$. En déduire un premier intervalle de confiance à 99% pour p, qui sera noté I_1 dans la suite.
5. Proposer une fonction Python `ic1()` fournissant (sous forme d'une liste de deux valeurs) un intervalle de confiance pour p au niveau 99%, à partir d'un échantillon fourni sous la forme d'une liste de 0 et de 1 (conformément au formalisme des X_i donné dans le préambule et dans la question **2(a)**).
6. En recourant à la seconde forme du théorème central limite, proposer un autre intervalle de confiance de niveau 99% pour p, qui sera noté I_2 dans la suite.
7. Comparer les intervalles de confiance I_1 et I_2 (lorsque n est suffisamment grand).
8. Le principe des intervalles de confiance est que si l'on dispose d'un grand nombre d'échantillons issus d'un tirage de Bernoulli de paramètre p et de longueur n (n grand) alors p doit appartenir, dans 99% des cas, aux intervalles de confiance I_1 et I_2. Vérifier ce fait par simulation.